THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 57, No. 10, December 1978
Printed in U.S.A.

Transaction Network, Telephones, and Terminals:

Transaction Network Operational Programs

By E. J. RODRIGUEZ

(Manuscript received June 6, 1978)

The Transaction Network Operational Programs provide the logic for switching data messages between terminals and Customer Service Centers. These programs also perform administrative and maintenance tasks. This paper describes program organization and various software functions.

I. INTRODUCTION

The stored program for the Transaction Network (TN) directs the operation of equipment that provides switched communications of short data messages between terminals and Customer Service Centers and between two different Customer Service Centers. The system is designed to meet operational requirements differing from those imposed on other stored program switching systems, such as the No. 1 Electronic Switching System (ESS).[1] Unlike such line-switched systems, a message-switched system provides no intrinsic end-to-end verification of the communication path or delivery of the message. The originating user relys on the system to properly deliver accepted messages to the appropriate destinations. This places stringent requirements on the message switching system to provide message protection, assurance of delivery, and privacy.

The TN stored program is described in three parts: the call processing programs which support the various service features and protocols, the maintenance programs which maintain an operational system in the presence of troubles and diagnose the faulty units, and the administrative programs which allow the telephone company to input office parameters and customer information into memory and to obtain traffic and maintenance measurements reports and billing information. This paper provides an overview of the various programs. Companion papers cover the hardware structure and service capabilities.

II. MESSAGE SWITCH

The message switching vehicle for TN is the 3A Processor. The controlling unit of the 3A complex is the 3A Central Control (3A CC), which is also used in the No. 2B ESS and No. 3 ESS installations. It is duplicated to provide continuous real-time operation with a high degree of system reliability. Attached to each 3A CC are a serial input/output channel (which serves low speed devices such as teletypewriters), several parallel input/output channels (which serve various input/output devices), and a Direct Memory Access (DMA) channel. Figure 1 is a diagram of the 3A Processor.

Basically, one 3A Processor always has active control over the system, while the other 3A Processor operates in a standby mode. Each 3A Processor has its own dedicated main memory. The on-line processor normally keeps both the on-line and stand-by memories up to date so that the standby processor can assume control of the system with an up-to-date storage area.

From a software point of view, the 3A Processor is supported by the Extended Operating System (EOS). This system consists of a set of program modules used by the TN programs to manage the effective use

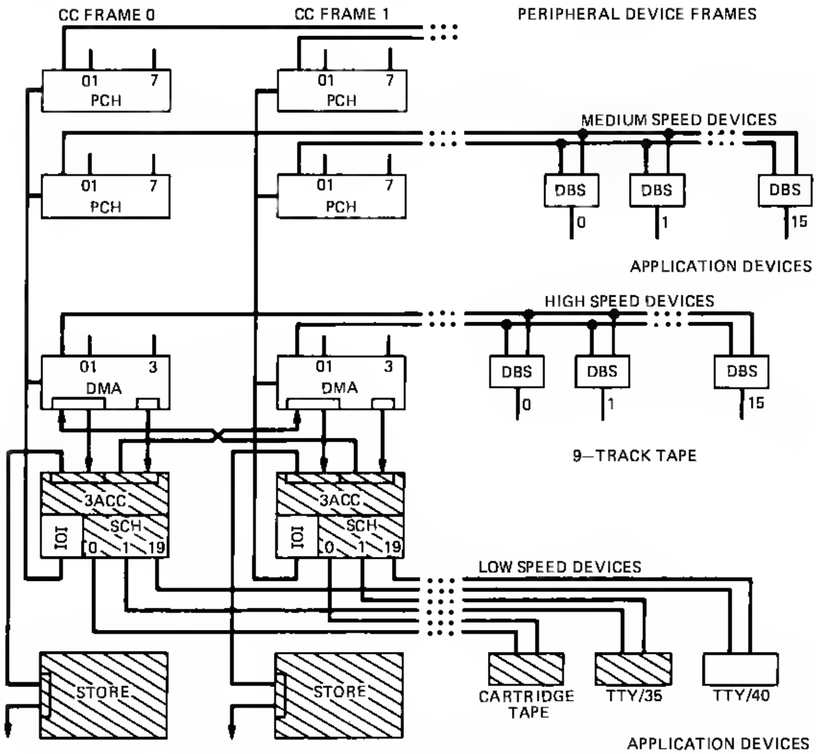


Fig. 1—3A Processor configuration.

of the processor resources and provide the basic maintenance philosophy of ESS.

EOS provides services such as timer control, event control, current-process control, interprocess communication, input/output control, and maintenance control. It also meets the stringent ESS service requirements by providing automatic reconfiguration, recovery phases, EOS audits, and processor diagnostics. These four elements enable the system to continue processing in the presence of 3A Processor hardware and software errors. In addition, EOS provides services to the TN software to successfully implement these elements as applicable for TN specific hardware and software.

III. SOFTWARE ORGANIZATION

The TN software can be divided into three categories: call processing, maintenance, and administration. It is made up of 43 cooperating asynchronous tasks* listed in Table I. Some of these tasks are executed on a scheduled basis using the timer facilities provided by EOS. The remaining tasks are executed upon demand using the interprocess communication facilities also found in EOS. Call processing tasks are assigned higher priorities than maintenance and administrative tasks.

Each task is allocated storage at system generation time, and all tasks residing in the system operate in a write-protected mode. Programs that are used infrequently (e.g., some administrative and maintenance type programs) reside on a cartridge tape and are brought into an overlay buffer in memory as the need arises.

The TN software (excluding EOS) consists of approximately 200,000 program store words. Table II illustrates the functional division of the software. Specific descriptions of the call processing, maintenance, and administration software are given in the following sections.

IV. CALL PROCESSING

The purposes of the TN call processing programs are to (*i*) respond rapidly in real time to the demands for service received from the polled, dial-in, and synchronous networks,[3] (*ii*) provide a large variety of service features, (*iii*) be reliable, and (*iv*) be capable of meeting various installation configurations. Basically, a data message received by the TN message switch passes through three stages of processing: (*i*) detection of a request for service, (*ii*) routing of the input message (along with validation of heading information) to a delivery queue, and (*iii*) servicing of the delivery queue and transmission of each message to its destination.

* A task is a computation that may be done concurrently with other computations (Ref. 2).

Table I — Transaction Network tasks in order of priority

Priority	Task Name
1	High Priority TN Initialization Task
2	Processor Switch Task
3	Synchronous Error Handler Task
4	Synchronous Time-Outs Task
4	Dial Line Adapter Driver Task
4	Synchronous Service Message Task
5	Audio Response Unit Driver Task
5	Dial Reallocation Slowdown Task
5	Polled Initialization Task
5	Synchronous Input Task
5	Synchronous Output Task
6	Audit Buffer and Recovery Task
7	Synchronous Recovery Task
8	AMA Recording
8	Dial Call Processing Message Task
8	Dial Protocol Timer Task
9	Polled Call Processing Background Task
10	Dialed Test Unit Scheduler
11	Polled Recovery Task
11	Memory Reallocation Monitor
11	Dialed Recovery Task
12	Dial Periodic Maintenance Task
13	Polled Periodic Maintenance Task
13	Switched Control and Monitor Task
14	System Status Panel Message Task
14	Input Message Handler Task
15	2-Second Status and Maintenance Traffic Scan Task
16	10-Second Status and Maintenance Traffic Scan Task
17	Quarter-Hourly Traffic Task
17	Diagnostic Message Handler Task
18	Hourly Traffic Task
19	Daily Traffic Task
19	Synchronous Recovery Task
20	Recent Changes and Verification Handler Task
21	ARU Loading Monitor
22	Maintenance 100-Second Scan
23	Synchronous Periodic Maintenance
24	Maintenance Hourly Task
25	Maintenance Daily Task
26	Cartridge Update Task
26	System Status Panel Task
27	Periodic Buffer Audit Task
28	Periodic Control Block Audit Task

Table II — Functional division of Transaction Network programs

Function	Percent
Call processing: polled, dial-in, synchronous	32
Maintenance: diagnostics, periodic, recovery	29
Administration—billing, traffic, recent changes, reallocation	29
Miscellaneous routines	10
	100%

4.1 Call processing concepts and definitions

Even though the polled, dial-in, and synchronous call processing tasks perform different functions and operate differently, some concepts followed by all call processing programs are basic. These are covered in the following sections.

4.1.1 Message format

A message consists of a heading field and a text field. The heading field is delimited by start of heading (SOH) and start of text (STX) characters, and it contains four items of information: (*i*) the called number, (*ii*) the calling number, (*iii*) the class of service character (CSCH), which identifies the type of service subscribed to by the polled and dial-in terminals and by the Customer Service Centers, and (*iv*) the message status field which indicates irregularities not covered by the data link protocols. The text field is delimited by the STX and end-of-text (ETX) characters. In some protocols, following the ETX is the Longitudinal Redundancy Check (LRC) character, which is used to detect possible transmission errors.

4.1.2 Data link protocols

Telephones, terminals, and Customer Service Center computers communicate with the message switch by following a protocol. A protocol is a detailed orderly procedure designed to ensure the successful transmission of messages from the origination to the destination points. Generally, it begins with a request for permission to transmit a message (bid). If the bid is accepted, then the message is transmitted, and if found acceptable by the destination, an acknowledgment (ACK) is returned to the originating station. The originating station concludes the transmission session by sending an end-of-transmission (EOT) sequence. If the message is not accepted, then a negative acknowledgment (NAK) is sent to the originating station, and error recovery procedures follow.

Presently, the TN call processing software supports five different protocols: three dial-in,[4] one polled,[5] and one synchronous.[6] Each protocol contains different message formats and error recovery procedures, as appropriate to the terminal capabilities and functions.

4.1.3 Buffers

As messages arrive in the message switch, they are temporarily stored in buffers. A common buffer pool serves requests from the polled, dial-in, and synchronous call processing programs. Any call processing program may request one or more message buffers at a time. Depending on the buffer utilization, the request may or may not be satisfied. If fewer buffers than requested are returned to the call processing programs, the buffer task indicates how many buffers are returned. Also, the buffer task maintains a register in memory which can be accessed at any time by the call processing programs indicating the buffer utilization. This number is used by call processing programs during peak periods to control the rate of message acceptance by the message switch.

4.1.4 Standard buffer format

Since messages arrive at the message switch with different heading and text fields (depending on the protocol), the call processing programs temporarily buffer all messages in a standard format. These buffers are in the Standard Buffer Format (SBF)—see Fig. 2. At this point, all messages found in the message switch are similar in structure.

4.1.5 Control blocks

Control blocks are dedicated areas of storage of varied size (from one word to several hundred words), which describe the characteristics of a customer service, a hardware device, or the state of the software. These blocks are created via teletypewriter commands, and their number depends on the size of the installation and the number of customers served by a particular message switch. Since there are various types of control blocks in the system, a directory is required to allow software access to the control blocks. This directory is called the Master Block Array (MBA). Figure 3 is a pictorial representation of the MBA.

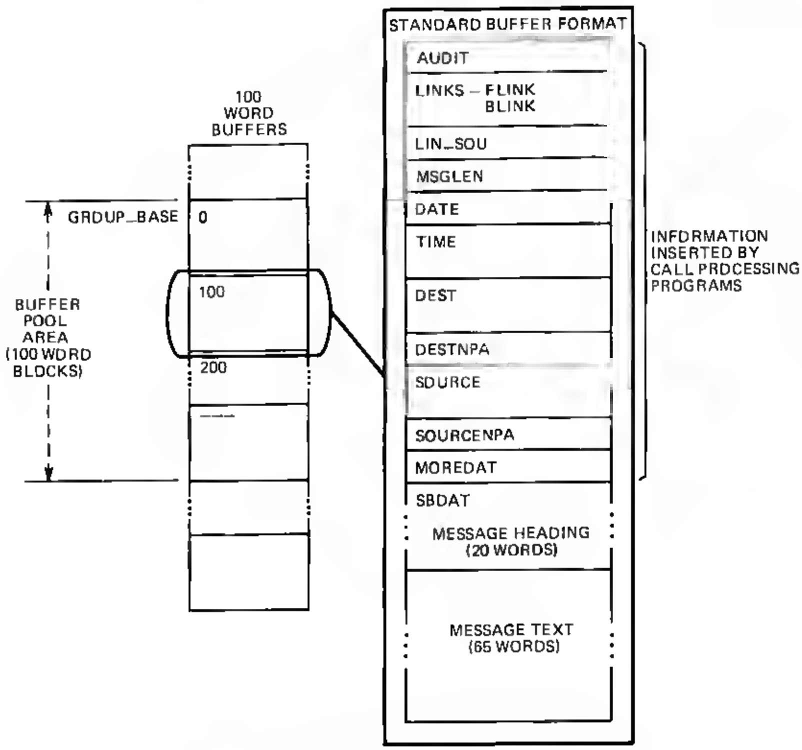


Fig. 2—Standard buffer format structure.

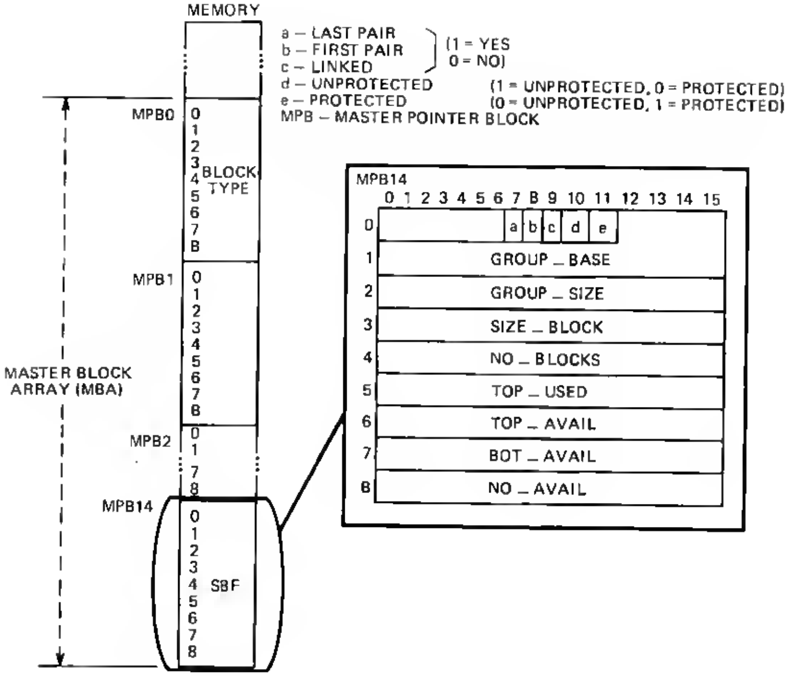


Fig. 3—Master block array.

4.1.6 Foreground and background tasks

All call processing programs serving the polled, dial-in, and synchronous networks follow the same type of organization. A foreground task performs the real-time operations (e.g., input and output), and a background task performs less real-time-sensitive operations (e.g., validity checks in the heading field, routing). Usually a background task is executed as a result of a request by a foreground task or another background task. A background task does not communicate directly with a foreground task.

4.1.7 Service messages

Service messages are messages between a polled terminal or the Customer Service Center and the message switch. They are used for testing purposes or to change the state of a synchronous line or group. A directory number of 0999 is assigned to the TN message switch to designate it for reception or origination of service messages.

4.2 Call processing overview

Figure 4 illustrates in general terms the steps followed by the call processing programs from reception of a message through its delivery. When the foreground call processing software detects activity from a TN peripheral device, it immediately requests a buffer from the buffer pool. Message characters are read one at a time and stored in the Standard Buffer Format. After the last character (ETX or LRC character) of the message is received, the protocol is completed and various validity and routing checks are performed. If no irregularities are found, then the address of the buffer containing the recently received message is sent via EOS to the background call processing task handling the delivery of the message. This task performs further validity checks, after which the foreground task transmits it to the destination. The foreground task completes the protocol and then awakens the billing task, via EOS, and sends it the buffer address. The billing task obtains the necessary information from the buffer to bill the message and releases the buffer back to the buffer pool.

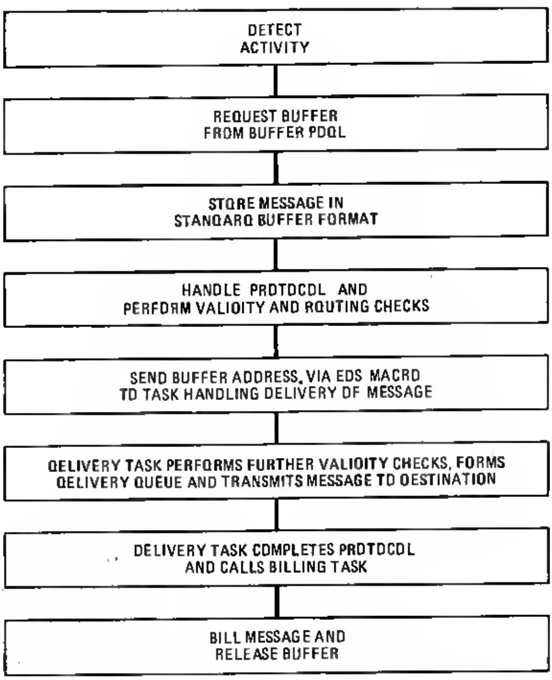


Fig. 4—Call processing overview.

4.3 Polled call processing

4.3.1 Polled access circuit

The basic building block of the polled network is the polled access circuit (PAC). It consists of dualized Data Station Selectors (DSSs) and Asynchronous Line Adapters (ALAs) served by two different Line Adapter Selectors (LASs), as shown in Fig. 5. In normal operation, half the terminals associated with a DSS are assigned to each ALA associated with that DSS. When a hardware unit (LAS, ALA, or DSS) or a transmission facility in one-half of a PAC is found to be defective, the unit or the facility is taken out of service and the terminals normally served on the defective half of the PAC are then served by the other half. This permits continued operation of all terminals in a PAC with somewhat slower service.

4.3.2 Polled control blocks

The polled call processing software controls the various polled lines and terminals via three different types of control blocks:

(*i*) The Asynchronous Line Controls Blocks (ALCBs), which contain all the information necessary to control a polled line. This includes such items as the state of the ALA, the state of one-half the PAC, the buffer address for a particular message, time-out indicators, the state of the protocol, traffic counters, retry counters, etc. There is one ALCB for each ALA, so that a PAC requires two ALCBs, one for each half of it.

(*ii*) The Terminal Control Blocks (TCBs), which contain the primary and secondary polling addresses to reach a terminal from either half of

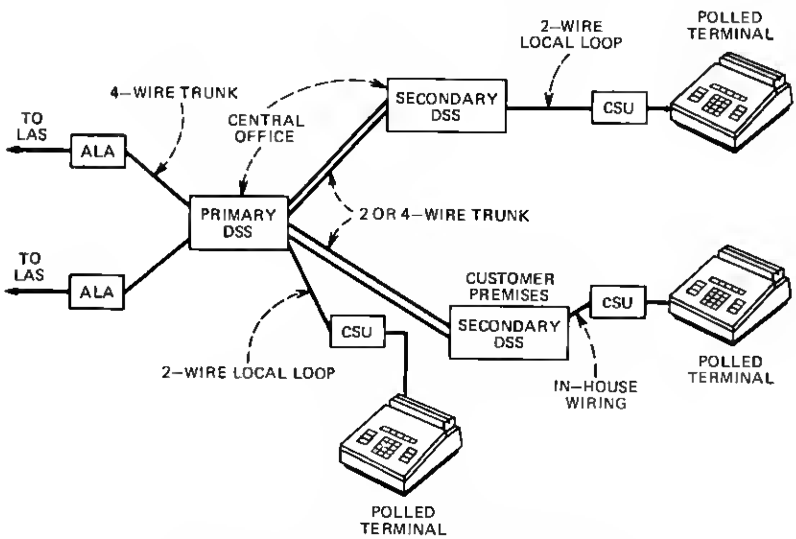


Fig. 5—Possible polled access circuit configuration.

a PAC, terminal options, and the number of the ALA serving the terminal. TCBs are arranged in memory by terminal number (from 1000 to 7999), but they are in linked lists according to polling order.

(*iii*) The Restricted Service Lists (RSLs), which contain up to ten Customer Service Center numbers that a restricted polled terminal may access.

4.3.3 Polled call processing operation

Polled call processing operates on a 50-ms scanning cycle. Every ALA is checked for activity on each cycle. This is done by first checking activity in the LASs serving the ALAs and, if activity is found, by then checking if one or more of the 16 ALAs served by an LAS indicates activity. An ALA indicates activity if it (*i*) needs polling addresses to be loaded, (*ii*) has received a message from a polled terminal, or (*iii*) has space available for output in its 64-character output buffer.

4.3.3.1 Polling. The ALA is capable of autonomously polling via an internal circulator. Polling addresses are loaded into the ALA from the message switch. After an initialization of the polled side hardware, the ALCBs are set to the POLLING state. This causes the LOAD POLLS routine to be called, which turns off the ALA circulation, outputs the polling addresses into the ALA by traversing the TCBs associated with it, and then turns on the circulator again.

Once the polls are loaded in the ALA and the circulator is turned on, the ALA transmits polls to each terminal without further direction from the message switch. Recirculation of the poll characters reduces the message switch work load. Interruption of the polling sequence occurs when a terminal begins transmission of an inquiry message or the message switch begins transmission of a response message to a terminal. In both cases, the ALA buffer containing the poll addresses is cleared.

4.3.3.2 Reading. When an ALA indicates activity and the RDA (receive data available) status bit is set, then the logical state of the line shifts to READING. As characters arrive in the ALA from the polled terminal, they are stored in its 64-character input buffer. The characters are then read by the message switch during each 50-ms scanning cycle, using a special microcoded communications instruction. This instruction stores the characters in a specified address, traps on special characters (e.g. SOH, STX, ETX), and computes the LRC sum. After the entire message is received and message format checks are made, the protocol is continued. An unsatisfactory message causes a NAK sequence to be sent to the terminal, and a finite number of retransmissions are attempted. A satisfactory message causes the program to send an ACK reply to the terminal. In either case, an EOT reply is expected from the terminal and the line state shifts to EOT-WAIT.

If the EOT reply is received after an ACK sequence has been sent to the

terminal, then the message is further checked and routed to its destination. Otherwise, it is discarded. In both cases, polling resumes. Before the message is sent to the task handling final delivery of the message, the following actions are taken by the polled call processing program:

(*i*) It verifies the format of the message heading to make sure field separators are found and the information is reasonable.

(*ii*) It checks whether the terminal is restricted or unrestricted. If restricted, the Restricted Service List (RSL) is checked to be sure it contains a valid Customer Service Center number. If unrestricted, it verifies that the Customer Service Center number is within range.

(*iii*) It checks the length of the message text so the TN 128-character limitation is not exceeded.

(*iv*) It fetches and stores the time and date.

(*v*) It converts the called and calling numbers into binary format.

In the event any errors are found, the message is returned to the polled terminal with an appropriate message status indicator.

4.3.3.3 Writing. A message sent by a Customer Service Center to a polled terminal is routed from the synchronous call processing background task, using EOS calls, to the polled background task. This task is normally in the "wait" state, and it is not executed until it is awakened by the synchronous background or polled foreground call processing tasks. The polled background task then checks the message for validity, makes sure the terminal addressed is in service, and that the line queue has not overflowed. If the message is to be delivered to a restricted polled terminal, it cross-checks the TCB and RSL to make sure the message route is authorized. If, while performing validity checks, the background task finds an irregularity, it then inserts a message status indicator and returns the message to the synchronous background call processing task. If all validity checks pass and if the line is in the POLLING state, the process is interrupted and the line state changes to WRITING. On the next 50-ms scan, the line state for the particular ALA is found in the WRITING mode, and the actual output of the message is then started. If the line is not in the POLLING state, the message is added to the line queue.

After the last character in the message (LRC) is transmitted, the line state is set to ACK-WAIT and then an ACK or NAK reply is expected from the terminal. At this point in the protocol, one of four things can happen: (*i*) the ACK reply is received, (*ii*) a NAK reply is received, (*iii*) something else is received, or (*iv*) nothing is received. Case (*i*) is the normal termination to the protocol, and the message is considered delivered. The billing task is then awakened, and polling is resumed on the line. In case (*ii*), the message is retransmitted on the same line one more time and then retried twice on the other half of the PAC before it is returned to the Customer Service Center. In cases (*iii*) and (*iv*), an ENQ (Enquiry) character is sent to the terminal up to three times to solicit the ACK reply.

Then in case (*iii*) the message is retried, at most twice on the main line and twice on the other half of the PAC before returning it. In case (*iv*), the message is retried only once on the other half of the PAC. Figure 6 illustrates the POLLINC, READING, and WRITINC process.

4.4 Dial call processing

4.4.1 Dial lines and protocols

The purpose of dial call processing is to process all transactions from the dial-in network using dial-in protocols. There are three protocols involved:

(*i*) Voice only: the simplest of the three. Inputs are *TOUCH-TONE*® characters, and output is automatic voice response only.

(*ii*) Voice/KAT: transmits automatic voice and/or Key Answer Tone (KAT) responses to the terminal. Inputs are *TOUCH-TONE* characters.

(*iii*) Data: transmits FSK responses to the terminal. Inputs are *TOUCH-TONE* characters.

In the dial-in network, a Line Adapter Selector (LAS) serves up to 16

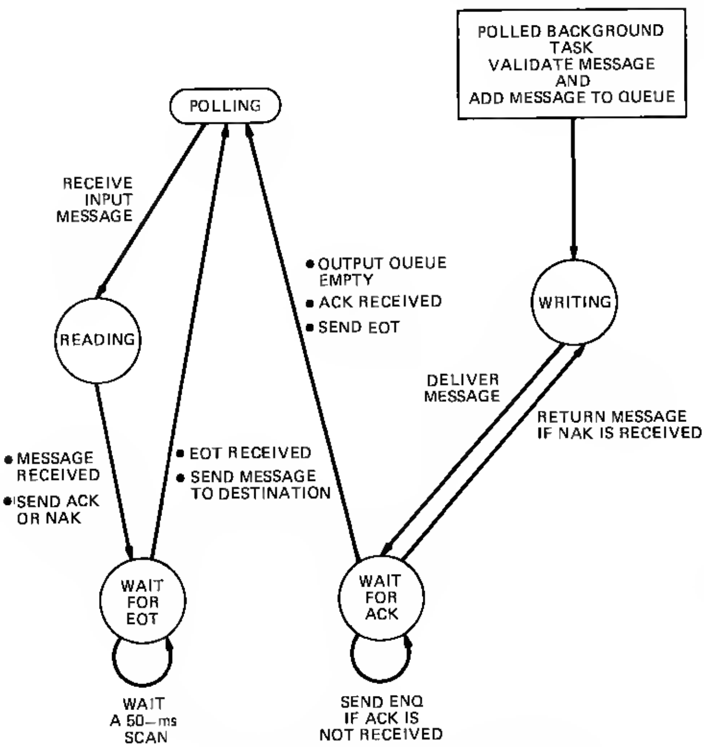


Fig. 6—Overview of polled network call processing.

Dial Line Adapters (DLAs). Each DLA, in turn, is connected to a 407A data set. The DLA serves as an interface among the 407A data set, the LAS, and the Audio Response Unit (ARU) which is used to automatically output the voice responses. The access lines to the 407A data sets appear in a line hunting group on a switching machine in the telephone network and are assigned a telephone number to access the message switch. To serve the various dial-in protocols, two different types of line hunting groups are supported. One serves the protocols which use voice responses. The other is dedicated to the FSK response protocol. Figure 7 shows the dial network configuration.

4.4.2 Dial control blocks

Dial call processing software controls the dial lines and the ARU via four different types of control blocks. These are:

(*i*) The Dial Line Control Blocks (DLCBs), which contain all the

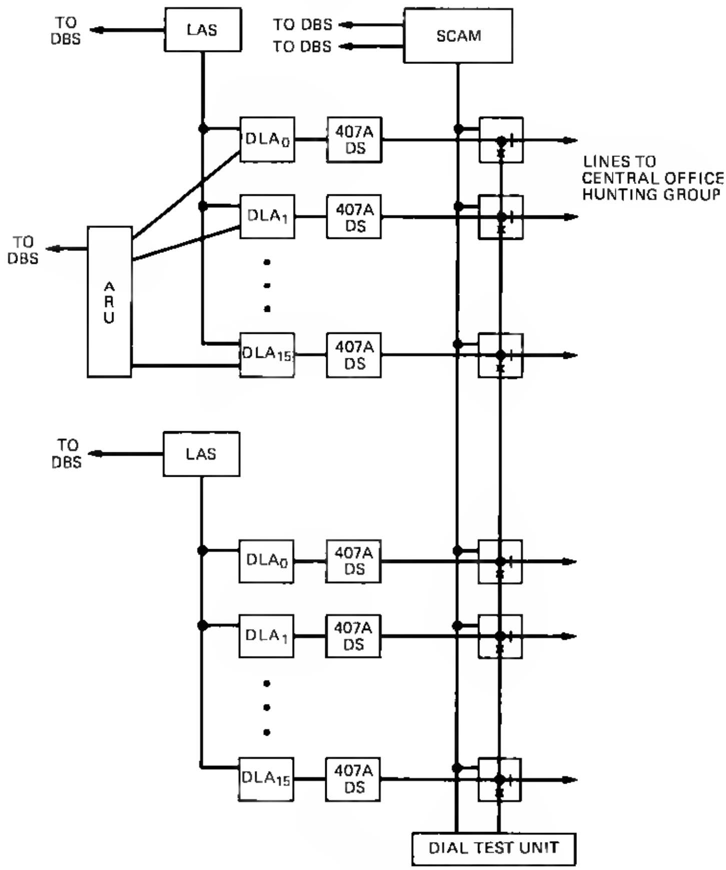


Fig. 7—Dial-in network configuration.

information needed to control the processing of a dial call. Items found in a DLCB are state of the dial line, pointer to the message buffer, state of the protocol, flags indicating if the ARU is producing speech on the line, etc.

(*ii*) The Line Hunting Group Control Blocks (LHGCBs), which define dial lines used for FSK responses or voice responses.

(*iii*) The Audio Response Unit Control Block (ARUCBs), which contain ARU status information and identify DLA ports connected to the ARU.

(*iv*) The speech list which contains the ARU memory addresses of the speech segments making up the ARU vocabulary.

4.4.3 Dial call processing operation

The dial call processing software (Fig. 8) consists of three tasks. Two tasks operate in the foreground environment, and they control input/output operations to the DLAs and ARU units; the other task operates in a background environment and handles timers and non-real time message processing operations.

4.4.3.1 Dial Line Adapter driver. The DLA driver operates on a scanning basis every 70 ms. Its basic function is to handle all DLA input/output operations.

Every scanning cycle, the DLA driver checks to see if any LASs serving DLAs have DLA service requests. If no LAS shows a request, then the DLA driver releases control until the next scan. Otherwise, the first LAS showing a service request is queried to see which DLAs are requesting service. If a DLA shows activity but is in a maintenance mode, it is skipped, and another DLA requesting service is checked. If none are found on that LAS, the next LAS showing a service request is examined.

A DLA service request consists of any of the following situations:

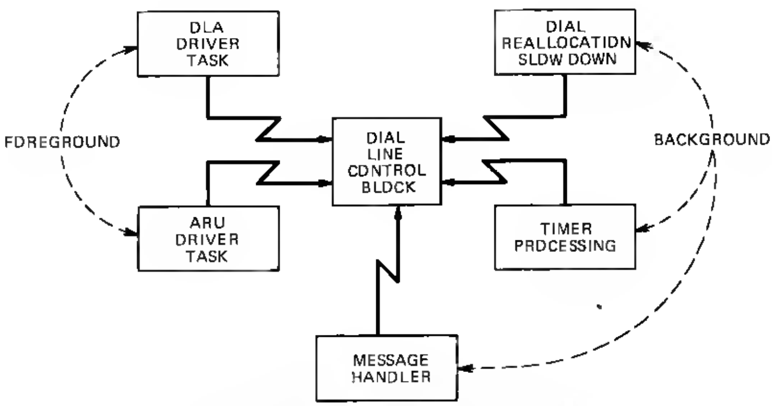


Fig. 8—Dial call processing—task communications.

(*i*) Line Ringing: the DLA driver answers the call and sends a 2025-Hz automatic answer signal on the line.
(*ii*) *TOUCH-TONE* input characters: the DLA driver processes the character and handles the protocol sequence.
(*iii*) FSK output characters: the DLA signals the message switch that it is ready for output.
(*iv*) Intercharacter timeouts: the DLA signals the message switch that the timeout between characters has elapsed. The protocol then will take appropriate action.
(*v*) Calling party disconnect: the DLA notifies the message switch of a line disconnect and awakens the billing task.

The DLA driver, in general, handles all protocol sequences. When the protocol is completed and if the message has passed all validity tests, the DLA driver sends the address of the buffer containing the message, via an EOS call, to the synchronous background call processing task.

4.4.3.2 Audio Response Unit (ARU) driver. The ARU is an output peripheral device used in TN to produce voice responses by piecing together digitized speech segments. It is operated from the message switch via an ARU driver. The ARU driver is executed every 100 ms, and its main purpose is to provide an ARU with addresses so speech can be generated in selected dial lines. Once speech has begun to be generated on a line, the ARU has to be given ARU memory addresses every 1/6 second. The ARU driver controls each line connected to it by referencing the DLCB. Silence is generated on any port whose DLA is not active or does not have a message to be sent to it.

4.4.3.3 Dial background message task. The dial background message task receives messages via EOS from the synchronous background call processing task. This task performs validity checks on the message. If irregularities are found, it returns the message to the synchronous background call processing task via EOS. Otherwise, it determines the protocol response type and either activates the ARU driver by setting a flag in the DLCB or causes activity on the DLA by initializing it. The DLA driver then will sense the service request and output the message.

The dial background task is also awakened when one of the several protocol timers elapses. Usually, a timeout causes a call to be disconnected.

4.5 Synchronous call processing

4.5.1 Synchronous line

Synchronous lines are used for communications with the Customer Service Center (CSC). Figure 9 illustrates the synchronous line arrangement. The synchronous line consists of a Synchronous Line Adapter (SLA), an analog data set or data service unit (DSU), and a dedicated line facility (analog or digital) which provides the communi-

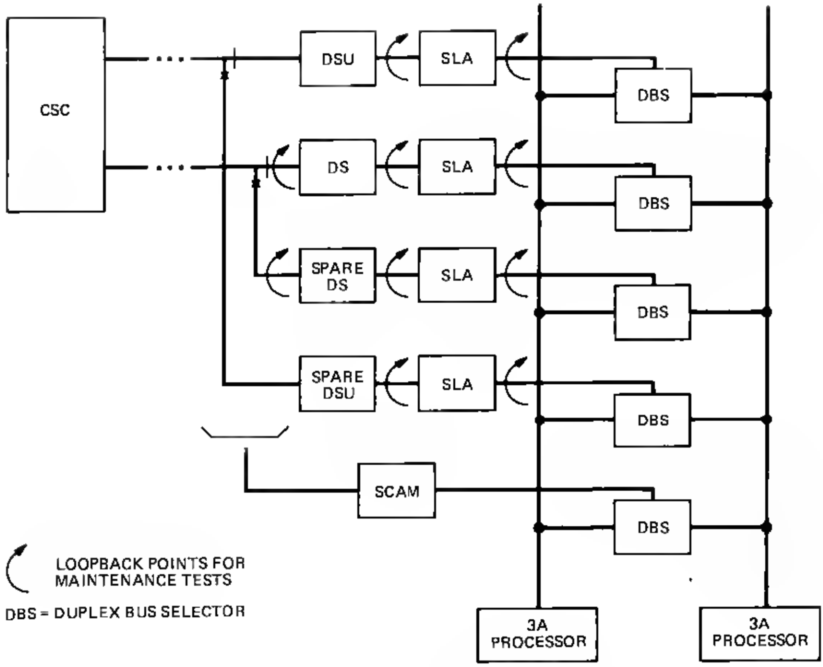


Fig. 9—Synchronous access line configurations.

cations link with the CSC. In addition, for reliability purposes, associated with synchronous lines of the same type (same speed, modem or DSU, and transmission facility type) is an SLA/data set or data service unit spare combination which can be switched in automatically if the normal synchronous port becomes defective. Figure 9 illustrates the synchronous line arrangement. Switching of the spare SLA/modem or DSU combination is through the Switch Control and Monitor (SCAM).

4.5.2 Synchronous groups and forwarding

A collection of synchronous lines communicating with a Customer Service Center (CSC) comprises a line "group." A CSC may subscribe to more than one line group. A line group is addressed as a single entity, but it may contain one or more synchronous lines. The synchronous call processing programs evenly distribute message traffic over all members of a line group.

The CSC may also select to have messages forwarded to an alternate CSC when it is inoperable or overloaded, or a line group is out of service.

4.5.3 Synchronous control blocks

Synchronous call processing software controls the various synchronous lines via two different types of control blocks:

(*i*) The Synchronous Line Control Blocks (SLCBs), which describe the characteristics of a synchronous line (speed, analog or digital, SLA spare) and the current state of the line and protocol, and contain pointers to the line message queues.

(*ii*) The Synchronous Group Control Blocks (SGCBs) which contain information about the lines making up the group, forwarding (or alternate delivery) CSC, pointer for message queues, etc.

4.5.4 Synchronous call processing operation

The synchronous call processing is interrupt driven. A foreground routine called the Synchronous Interrupt Service Routine (SISR) services all interrupts issued by the various SLAs by saving the current state of the 3A Processor, enabling higher priority interrupts, identifying the interrupting SLA, and transferring control to the appropriate state driver. After the state dependent driver executes, the SISR determines if any other SLAs require service. If so, they are serviced; if not, the interrupt is cleared and the prior 3A Processor state is restored, allowing lower priority tasks to execute.

4.5.4.1 State dependent drivers. The state dependent drivers can be classified into four categories: (*i*) the protocol driver which controls the processing of communications with CSCs, (*ii*) the diagnostic driver which controls the execution of SLA diagnostics, (*iii*) the test driver which is used when periodic tests are performed on the SLA, and (*iv*) the clear interrupt driver which is used when a faulty SLA is suspected and all interrupts from it are to be ignored.

4.5.4.2 Protocol driver. The protocol driver, which presently supports binary synchronous communications procedures, is in one of three states:

(*i*) The control state indicates that the line is idle. In this state, either the TN message switch or the CSC can initiate a request.

(*ii*) The receive state indicates that the message switch is receiving from the CSC.

(*iii*) The transmit state indicates that the message switch is transmitting to a CSC.

Within a receive or transmit state, substates are defined that describe the exact position of a message transmission within the protocol. The protocol driver decodes the current state and sub-state, handles the data transmission or reception based on the allowed actions in the current state, and advances to a new state as needed.

The transmit state handles the grouping of messages into records and

records into blocks. It also handles many of the allowed transmit options for synchronous lines. Some of these options are: record size, block size, number of records per block, number of blocks per transmission, message prefix, message suffix, optional heading control characters and separators, and deletion or insertion of SOH on intermediate records within a block.

The receive state separates blocks into records and records into messages. It also handles many of the allowed received options for a synchronous line. These options are similar to the transmit options previously listed.

4.5.4.3 Message queues. The interface between the Synchronous Interrupt Service Routine (SISR) and the synchronous background tasks is via various queues and EOS events. The queues depend upon the direction of the message and whether the message is a data message or a service message.

Messages to be transmitted to a CSC are placed in a group queue by the synchronous call processing background task. The background task looks for a line in the group that is in the control state. If a line is found, it is initiated by executing (from the background task) a command to the SLA to force it to cause an interrupt. The SISR then checks the group queue, moves the message into a queue, transmits the message to the CSC and, if successful delivery of the message is accomplished, awakens the billing task. If no synchronous lines are found in the control state, this means that all lines are being used. Before returning a line to the control state, the group queue is checked and if any messages are found, they are delivered. In either case, service messages are given priority of delivery over data messages.

Messages received from a CSC are first placed in an input line queue. If the message is received correctly, it is then moved to a synchronous input queue if it is a data message or to a service message input queue if it is a service message. These are special queues which are not associated with a group or line since the routing information has not been decoded at this time. The synchronous background task is informed of queue entries by an EOS event.

4.5.4.4 Synchronous background receive task. The synchronous background receive task retrieves messages from the synchronous input queue and converts them into the Standard Buffer Format (SBF). This involves moving the various items of the heading into SBF specified locations and inserting the heading field, record, and group separators.

The background receive task then proceeds to check such things as the message sequence number, the message status characters, and the called and calling number fields. A message is returned to the sender if it fails any of the preceding tests. A check is made to determine if the specified calling number is eitber the true calling number or the number of the group forwarded to by the specified calling number. If the calling

number is invalid, the message is returned. As described below, all screening of messages is based on the options for the specified calling number as opposed to the actual (alternate) calling number.

If the called number is a group number (i.e., a CSC-to-CSC call), a check is made of the Class of Service Character (CSCH) to determine if it is appropriate for CSC-to-CSC communications. If this or other tests fail, the message is returned. Assuming a valid called number, the message is then routed to either the polled or dial background task or to the synchronous background transmit task.

4.5.4.5 Synchronous background transmit task. Through EOS, the synchronous background transmit task receives messages from the polled, dial-in, and synchronous call processing programs.

The message status characters of the received message are examined to determine if the message is being returned to the CSC because it cannot be delivered to the called number or if the message is a data message intended for the CSC. If the message is being returned, an attempt is made to deliver the message to the group that originally sent the message. This group may not be the group identified as the destination in the message heading because the initial inquiry may have been forwarded.

Based on the destination group number and the synchronous group control block, the calling number and Class of Service Character are screened. The calling number identifies what type of terminal (polled, dial, or another CSC) originated the message and the Class of Service Character identifies the type of call (unrestricted, restricted, etc). If the message fails any part of the screening, the message is returned to the sender with the appropriate message status characters included on the message heading.

After a message passes screening, the state of the called group and the length of the called group queue are examined. If tbe called group cannot accept the message, an attempt is made to forward tbe message to another group, provided the called group has a forwarding point specified in its SGCB. Once a destination group has been determined, the message is added to the group message queue for the destination group. The SISR then removes messages from the group queue and transmits them to a CSC.

4.5.4.6 Service messages. As previously mentioned, service messages are used to coordinate CSC and TN activities for a synchronous line group and the lines in the group. A special service message task handles all service messages.

Processing the service message heading is similar to processing data message headings. Processing the service message text depends upon the type of service message. The service message types are SET STATE REQUEST, SET STATE ACK, REPORT STATE REQUEST, REPORT STATE ACK, HALT WAIT REQUEST, HALT WAIT ACK, ECHO REQUEST, and

ECHO ACK. Request messages are accepted only if the TN has not issued an unanswered request and thus is not waiting for an acknowledgment message (e.g., SET STATE ACK). Unaccepted request messages are returned to the CSC. The exceptions to this are a HALT WAIT REQUEST and a REPORT STATE REQUEST. A HALT WAIT REQUEST is accepted at any time. This request serves as an acknowledgment to all outstanding requests. A REPORT STATE REQUEST serves as a signal for the TN to repeat the last request if the REPORT STATE REQUEST is received while the TN is waiting for an acknowledgment.

Acknowledgment messages are accepted only if a request of the same type is outstanding and the service message sequence number of the acknowledgment is the same as the sequence number transmitted with the request. Unaccepted acknowledgments are returned to the CSC. The processing of a service message request requires the generation of an acknowledgment for each request received.

Service message requests from the TN to the CSC are initiated by maintenance and recovery tasks. These tasks pass a message to the service message task describing the type of service message they want transmitted to the CSC. The service message task then creates the complete service message and coordinates its transmission to the CSC. If the message is a SET STATE REQUEST for one line or more, then the service message task will determine if the group state should be changed. If it should, a group SET STATE REQUEST is also generated and transmitted to the CSC.

The service message task will automatically send a HALT WAIT REQUEST if a TN-initiated request is not acknowledged within one minute. Following the receipt of the acknowledgment to the HALT WAIT (or after another minute), the original request is repeated. If this second request is not acknowledged within one minute, the synchronous recovery program is invoked for the line over which the requests were sent.

4.6 Billing

The billing task is usually in the "wait" state, and it is awakened by the polled, dial-in, and synchronous call processing programs whenever a message has been successfully delivered to the destination point. For example, the call to the billing task for a message originated by a polled terminal is done by the synchronous call processing task after it was successfully transmitted to the Customer Service Center (see Fig. 4).

When the billing task is awakened, the address of the buffer containing a message in the Standard Buffer Format (SBF) is passed to it. The billing task then obtains from the SBF items such as the called and calling numbers, the number of characters in the text of the message, message status information, and the connection time. It formats this information into records compatible with the Automatic Message Accounting (AMA)

standards and then writes the records on a 1600-b/in. 9-track magnetic tape.

Once tbe billing task has obtained all the information necessary from the SBF, it calls the buffer task which releases the buffer to the common buffer pool. At this point, the processing related to the delivery of a message terminates.

The 9-track magnetic tape is generated from a *Pro*grammed *Ma*gnetic *T*ape *S*ystem (PROMATS). Duplicate PROMATS drives are used for reliability purposes. Therefore, another function of the billing task is to control the operation of PROMATS.

V. GENERAL ADMINISTRATION AND MAINTENANCE PLANS

The dependability, administration, and maintainability objectives, when applied to stored program switching systems, define the need, in computer programming terms, for an on-line, real-time, high-availability machine.[7] This requires careful initial systems planning in basic redundancy configurations, in the human interface to the machine, and in hardware-software tradeoffs. Approximately two-thirds of the total TN software is dedicated to maintenance and administrative programs that are used to manage system redundancy, control diagnostic routines, make performance measurements, and provide communications with the craftsperson. It is the need to keep the message switch operational during periods of growth and change of customer services, the need to maintain calls in progress during switches to standby equipment, and the requirement for providing simultaneous on-line communications with a number of craftspersons that adds extensively to the program structure and makes maintenance more than simply a matter of diagnostics.

5.1 Operator interface to message switch

The major communications vehicle between the message switch and the craftsperson is by teletypewriter. In addition, audible alarms and visual displays are used to alert the craftsperson to trouble conditions which are subsequently more fully reported on a teletypewriter. Manual controls are also available for taking restart action when the system has lost its "sanity" to the point where it can no longer interpret teletypewriter input commands.

5.1.1 Teletypewriter facilities

A typical TN installation will include four teletypewriter facilities:

(*i*) Maintenance: This TTY reports all system maintenance activity (troubles detected, diagnostic results, maintenance and traffic registers) and accepts all system input messages (maintenance and other).

(*ii*) Service Order: This TTY is used to create tables in memory to reflect changes in customer information (directory numbers, features, billing arrangements, etc.).

(*iii*) Traffic: This TTY provides traffic data according to defined schedules. Specific data can be requested and the schedule can be changed by input messages.

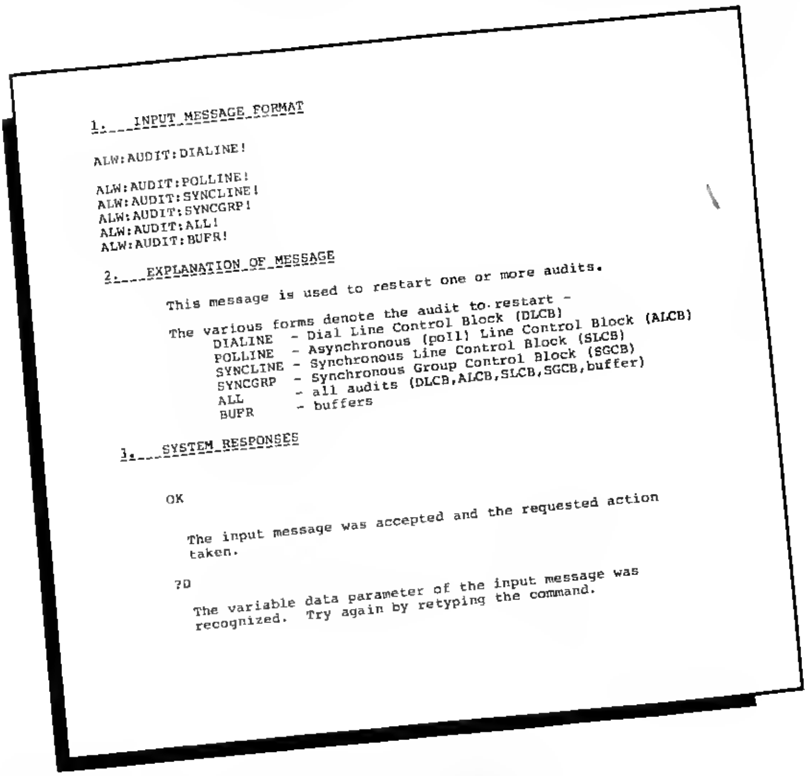
(*iv*) Repair: This TTY reports equipment failures in a remote telephone company service bureau.

5.1.2 Documentation

The human interface to the message switch is built on a hierarchy of documents with which the craftsperson must be familiar.

The Input Message (IM) and Output Message (OM) Manuals define all possible teletypewriter messages which are programmed into the machine and lists all acceptable input requests and the expected response to them. Figure 10 shows a typical input message entry.

When the output message gives specific diagnostic data, the OM points to a Trouble Locating Manual (TLM) which provides a description of



```
1.   INPUT MESSAGE FORMAT

ALW:AUDIT:DIALINE!

ALW:AUDIT:POLLINE!
ALW:AUDIT:SYNCLINE!
ALW:AUDIT:SYNCGRP!
ALW:AUDIT:ALL!
ALW:AUDIT:BUFR!

2.   EXPLANATION OF MESSAGE

     This message is used to restart one or more audits.

     The various forms denote the audit to restart -
          DIALINE   - Dial Line Control Block (DLCB)
          POLLINE   - Asynchronous (poll) Line Control Block (ALCB)
          SYNCLINE  - Synchronous Line Control Block (SLCB)
          SYNCGRP   - Synchronous Group Control Block (SGCB)
          ALL       - all audits (DLCB,ALCB,SLCB,SGCB,buffer)
          BUFR      - buffers

3.   SYSTEM RESPONSES

     OK

       The input message was accepted and the requested action
       taken.

     ?D

       The variable data parameter of the input message was
       recognized.  Try again by retyping the command.
```

Fig. 10—Audit input message entry.

the trouble number printed in the output message and indicates a specific set of suspected circuit packs. Should this information prove inadequate (that is, replacement of packs does not clear the trouble), repair procedures might then involve reference to the more basic maintenance documents, including program listings and schematic drawings.

A series of internal Bell System documents called Bell System Practices (BSPs) including Task Oriented Practices (TOPS) is also provided as basic training documents. These documents give overall system descriptions in addition to detailed operational and administrative procedures to be followed by the craftsperson. They contain extensive cross references.

VI. MAINTENANCE PROGRAMS

Every TN peripheral hardware device has features to facilitate maintenance of the network from the message switch. In addition, reliability of service is achieved by providing:

(*i*) Dualized line adapter transmission facilities and Data Station Selectors (DSSs) in the polled network.

(*ii*) Multiple dial-in ports in central office line hunting groups for terminals which access the message switch by way of the dial network.

(*iii*) A way of sparing Synchronous Line Adapters and modems or data service units of a particular speed and type with one backup. Therefore, the number of spares depends on line speeds (2400, 4800, or 9600 b/s) and the type of channels (analog or digital) involved.

The actual process of maintaining TN (from a software point of view) consists of four areas: *trouble detection, system recovery, diagnostics*, and *service verification*. Trouble detection is the process of recognizing the existence of a hardware error. Recovery is the process of bypassing a hardware error, usually by switching to a new unit or initiating other software corrective action. Diagnostics is the process of isolating a fault to a particular device or circuit board. Service verification procedures allow for automatic rechecks on service restoral. These four areas of maintenance are discussed in Sections 6.1 to 6.4.

6.1 Trouble detection

Four techniques are used to detect errors: self-checking hardware, periodic testing, audits, and protocols. The self-checking hardware usually consists of encoding and decoding instructions and data words into an m-out-of-n code or a parity code. Periodic testing is the process of executing software tests on hardware that is not self-checking. Audit programs protect the TN software from the effects of data mutilation

by detecting and correcting errors in the various control blocks and message buffers. Protocols are used to transmit and receive data from a remote station. Failures in the protocol indicate possible hardware errors. Protocol checks include block cbeck character codes such as the Longitudinal Redundancy Cbeck (LRC), message lengtb errors, format errors, or time-out errors. In addition, detection of excessive transient error counts in a hardware device may indicate a trouble condition. A transient error is defined as an incorrect operation of a hardware device which does not reoccur on a subsequent retry of the same operation.

6.1.1 Hardware checks

Three hardware methods are available for indicating faults to the 3A Processor: hardware initialization, interrupts, and status information. Hardware initialization is tbe process of initializing hardware registers to a predefined state and passing control to a particular software routine whicb can analyze the reason for the initialization and take appropriate action. Certain less severe faults cause interrupts to the processor rather than a hardware initialization. For example, faults in the serial channel or the memory on the off-line processor cause interrupts. The final method of indicating faults is status information. The processor hardware maintains status registers on the state of the processor and peripheral devices. Also, success/failure status is provided at the completion of every input/output (I/O) instruction which belps detect hardware and/or transient faults.

6.1.2 Periodic maintenance

Periodic maintenance programs, as the name implies, run on a periodic basis. They attempt to detect hardware faults. In general, periodic maintenance programs perform loopback tests on the various devices. The following examples illustrate how periodic maintenance is performed:

(*i*) The synchronous periodic maintenance programs test that every line has been active during a specified time interval. If a line had been idle for a full time interval (e.g., last 5 minutes), a service message is generated to test the line. If this service message fails to reach the Customer Service Center computer, recovery routines are automatically invoked.

(*ii*) In the polled network, the test message generator associated with a DSS is polled on a periodic time interval (e.g., every 10 minutes). This causes an 11-word message to be returned to the Message Switch whicb contains status information on polled terminal loops and the DSS power supply. A valid test message generator response performs two functions: (*a*) it allows the polled periodic maintenance programs to verify that the various hardware ele-

ments of the polled network (DBS, LAS, ALAs, transmission facilities) are performing properly, (*b*) it allows the periodic maintenance programs to analyze the loop current of all active polled terminals on that polled line.

6.1.3 Audits

The TN programs depend heavily on data stored in the various control blocks to record the states of messages and of system hardware and software resources. Hardware errors, program bugs, and incorrect manual operations can mutilate data in the various control blocks, causing messages to be mishandled and leaving system resources in unusable states. In addition, data errors could propagate throughout the call store data causing service to degrade, possibly creating the need for a system initialization (Section 6.2) to recover from errors.

Some of these errors are eliminated by defensive programming techniques. However, some types of errors would require a prohibitive amount of processor time to prevent, and still other more subtle errors could not be readily found using defensive programming techniques. Hence, audit programs are needed to protect the Transaction Network software from the effects of data multilation. These programs detect and correct errors in the various control blocks and message buffers.

6.1.3.1 Memory partitions. Memory is partitioned into two general regions by the 3A Processor hardware: write protected regions and read/write regions. Write protected regions are subdivided into two classes. Class I is always write-protected and contains such items as programs and constants. Class II is usually write-protected and contains data areas that change infrequently. Read/write regions contain transient data areas that change frequently due to call processing or maintenance actions and are defined as Class III regions. Audit programs are written to protect the read/write regions.

6.1.3.2 Types of audits. Audits are performed mainly on the polled, dial-in, and synchronous control blocks and on the buffers used in the system to temporarily store messages. The type of audits performed depend on the format and contents of the control block or buffer. The various audits performed are described in the next five paragraphs.

Linked List Audit. Control blocks are usually linked using a circular, double-linked list. Each element in the list consists of a data field and a header. The header consists of three fields: the forward link pointer which points to the next element in the list, the backward link pointer which points to the previous element in the list, and the audit word which contains a start-of-list bit, an end-of-list bit, a middle-of-list bit, and an element-type field. These bits indicate if the control block is the first, last, or middle in the list. Audit programs can use the properties of linked lists to verify that a control block belongs to a particular list and that the

control blocks are interconnected properly. Audit programs can also check the audit word to verify the validity of control blocks. Correction of a bad control block interconnection can also be accomplished by using the forward pointer, the backward pointer, or the element type.

Status Field Audits. Associated with each piece of equipment in the TN are status fields which describe the current state of the equipment. One status field describes whether the equipment is active, standby, out-of-service, or unavailable. Another field, for example, describes whether data are being transmitted or received. An error in a status field could have unpredictable results. Therefore, all status fields are encoded into an m-out-of-n code or a parity code. For example, a 1-out-of-4 code could be used to describe active, standby, out-of-service, or unavailable conditions. By encoding the various status fields, audit programs can verify the correctness of the code.

Message Buffer Audits. Message buffers residing temporarily in the Message Switch are in the so-called Standard Buffer Format (SBF). The SBF is divided in two parts, the heading and the text. The heading follows a predetermined format and has a certain number of fields. Therefore, audit programs can quickly verify any violation of the heading format. A Longitudinal Redundancy Cbeck (LRC) character is calculated for the text portion of the SBF and stored as the last entry. Audit programs calculate the LRC character for the text portion of the SBF and compare it with the already existing LRC character in the SBF to verify that the text portion of the SBF has not been overwritten.

Consistency Checks Audits. Certain other checks are made by the audit programs on transient data areas by examining the properties of the data. These checks consist of checking data words for a minimum value, a maximum value, a finite set of values, a unique value, a common value, a redundant value, or certain bits which are always zeros or ones.

Timeout Audits. Timeout audits are performed on message buffers. These audits are performed every 5 minutes. If the same message is found in a buffer during the execution of two timeout audits, this buffer is released back to the common buffer pool.

6.1.3.3 Audit control. Audit programs run as background tasks with a low priority status. Most audit programs can be initiated/inhibited via teletypewriter request. An audit failure is reported via a message printed on the maintenance teletypewriter.

6.2 System recovery

In a complex program-controlled system such as TN, hardware or software malfunctions can occur which result in improper call processing actions. The purpose of recovery software is to respond to a report from a call processing program or from a maintenance program of a software

or suspected hardware fault. In the case of a suspected hardware fault, the software will either confirm that the fault exists, or it will dismiss the report. If recovery software recognizes a device as being faulty (see Section 6.2.1), it will report the trouble via a teletypewriter message so that craft personnel can take appropriate repair actions. Alarms and system status panel lamps are also activated. Communication line(s) associated with the faulty hardware are removed from service, reconfiguration is attempted (for example, if a polled line is removed from service, its traffic load is assumed by the other half of a polled access circuit), and call processing is alerted of the present hardware configuration. If the fault prevents access to a Duplex Bus Selector (DBS), a processor switch is attempted.

In the case of software faults, correction techniques such as audits are applied. Occassionally, however, problems arise which are serious enough that severe recovery action, known as initialization (see Section 6.2.2), is necessary.

6.2.1 Fault Isolation

Isolation of a faulty device in the TN is based upon a multistep test process. When recovery programs are notified of suspected trouble in the polled, dial-in, or synchronous networks, the specific line or circuit number, but not the faulty unit, is identified. For example, a polled line includes a Duplex Bus Selector (DBS), Line Adapter Selector (LAS), Asynchronous Line Adapter (ALA), Data Station Selectors (DSSs) and interoffice facilities between a DSS and ALA and between two DSSs. Isolation of a fault in a polled line consists of performing loopbacks, as shown in Fig. 11, at the various network points until a failure is detected.

6.2.2 Initialization

The severity of an initialization determines the degree to which service is disrupted. Seven labels of severity or phases are provided so that increasingly drastic initialization actions can be performed until proper operation is resumed. This is determined by letting the system run for about 90 seconds in a particular initialization level.

6.2.2.1 Initialization levels. The initialization levels represent a compromise between maximizing speed of recovery and minimizing disruption of TN service. Seven levels are provided:

(*i*) Level 1: Only the task running at the time of initialization is restarted. An attempt is made by the task to restart the line or group doing input/output operations when the initialization occurred. This is done by referencing the registers as they were before the initialization. All messages being received on that line are aborted, and all messages destined for the line are rerouted to it.

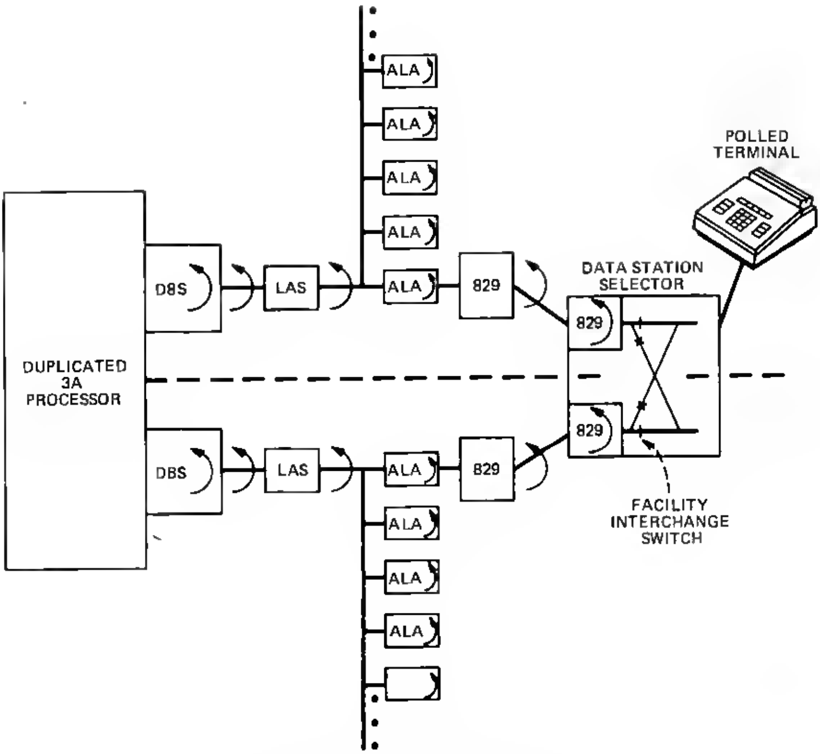


Fig. 11—Asynchronous polled access line loopbacks (↻).

(*ii*) Level 2: Only the running task is restarted. But now, since Level 1 obviously failed, it is necessary to do something more drastic. All line and group control blocks handled by the running task are audited, and those which fail audits are initialized to a known state. The messages that had been destined for those lines are then rerouted to them.

(*iii*) Level 3: This is the first level at which all tasks in the system are restarted. An initialization similar to Level 2 is executed for each task. The major difference is that queue pointers in the line and group control blocks are treated as special cases. If the line control block fails the audit, then the queue pointers are audited. If they fail, then the system is initialized to the next level. Otherwise, only the line control block which failed audits and the associated hardware are initialized.

(*iv*) Level 4: At this level a last effort is made to recover system sanity before all transient data are reinitialized. All line and group control blocks are reconstructed from write-protected data, with the exception of the queue pointers. They are audited and, if the audit fails, the level is escalated.

(*v*) Level 5: For Levels 5 and 6, the EOS will reinitialize all transient data. For Level 5, each task will initialize all transient data (line control blocks, queues, etc.). All the message buffers will then be audited. Those which pass audits will be routed to their appropriate destination. All messages in progress will be lost.

(*vi*) Level 6: All transient data are reinitialized. All hardware is reinitialized. All message buffers are relinked onto the available list. All messages in the system are lost.

(*vii*) Level 7: If Level 6 initialization also fails, then a bootstrap occurs and all programs and data areas are reloaded from the cartridge tape.

6.3 Diagnostics

The objective of diagnostics programs is to produce a teletypewriter printout which isolates a fault to as few circuit packs, cables, power units, and wiring areas or installation options as possible. In TN, diagnostics are only executed in response to a teletypewriter input message. A failure in a diagnostic is reported by a trouble number. This trouble number is used as an index into the Trouble Locating Manual (see Fig. 12), where a description of the failed test is found, along with important cautions and comments and the list of suspected circuit packs.

The diagnostic programs are also used for restoring equipment to service after repair and for testing new equipment additions. A new piece of equipment is not allowed into service until diagnostics pass all tests.

Although there are many common elements among the several TN peripheral unit diagnostics, each device diagnostic must be intimately tailored to the design of the hardware unit being diagnosed. To accommodate this situation, the diagnostics have been designed in a table-driven structure; a unique table exists for each hardware device being diagnosed. All diagnostic programs are brought into an overlay buffer in memory from the cartridge tape as requested.

6.4 Service verification: alarms

Teletypewriter output messages which report trouble conditions are assigned one of three alarm levels: critical, major, or minor. These levels are represented by printing *C, **, or * prior to the output message. Also, audible alarms and lamp indicators are activated according to the output message alarm level.

```
TRANSACTION NETWORK PERIPHERAL UNIT TROUBLE LOCATING MANUAL      TRBL 0402

                                                  PROG DSSDGN    0402

 0402   TEST  2, DEVICE: DSS (PRIMARY)

AN ATTEMPT TO ACCESS THE DSS BEING TESTED VIA LINE A (THE LINE
CONNECTED TO THE LOWER NUMBERED LINE SERVING THIS DSS) HAS FAILED.
THE TEST MESSAGE GENERATOR WAS POLLED, AND AN IMPROPER RESPONSE WAS
OBTAINED.  IF BIT 0 OF THE PATTERN PRINTED IS SET RERUN THE TEST
OTHERWISE THE DATA WORD PRINTED OUT HAS THE FOLLOWING
INTERPRETATION:

BIT:  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 |
=============================================|
      1 X X X X X X X X X  X  X  X  X  X  X  | SEE NOTE 1 BELOW
      0 0 0 0 0 0 0 0 0 0  0  0  0  1  0  0  | SEE NOTE 2 BELOW
      0 0 0 0 0 0 0 0 0 0  1  0  0  0  0  0  | SEE NOTE 3 BELOW
      0 0 0 0 0 0 0 0 0 1  0  0  1  1  1  1  | SEE NOTE 4 BELOW

NOTE 1:
 A PARITY ERROR HAS PROBABLY OCCURRED IN TRANSMISSION FROM THE DSS.
 RERUN THIS TEST.  IF THIS SYMPTOM PERSISTS, A FAULTY LINE IS
 POSSIBLE, OR A FAULTY RF1A CIRCUIT PACK IS POSSIBLE.  RUNNING THE
 LINE INTERCHANGE TEST (TEST6) MAY HELP TO RESOLVE THIS.

NOTE 2:
 POWER SUPPLY A IS FAULTY.  REPLACE CIRCUIT PACK RG1A.

NOTE 3:
 POWER SUPPLY B IS FAULTY.  REPLACE CIRCUIT PACK RG1B.

NOTE 4:
 RF1(A), RF2(A), RF4(1) OR TRUNK A IS SUSPECTED.
                           ********************
```

Fig. 12—Trouble locating manual description for trouble number 402.

VII. ADMINISTRATION PROGRAMS

7.1 Background

Administrative functions for the Transaction Network (TN) may be loosely categorized as either those dealing with system or customer changes and growth or those necessary for ongoing operations and system evaluation. Reallocations and service order changes clearly belong in the first group, while maintenance and traffic measurements fit into the second.

Basically, the software administrative functions for TN are:

(*i*) *Reallocations.* Reallocation programs are designed to deal with major network office and customer equipment growth, and are also used for general allocation of memory whenever new storage is added to the system. One example is the case in which the number of previously allocated control blocks are nearing a high percentage of usage. Consequently, new storage must be allocated to the block concerned. Since it is required that all blocks of a certain type occupy contiguous memory, overlapping could occur which tben requires a rearrangement of all

blocks concerned. The reallocation procedure is intended to be performed much less often than the service order cbanges and is inherently more complex.

(ii) *Service Order Changes*. Service order change programs are required to provide a craftsperson or service order clerk with the means to update system memory to reflect changes which are subscriber-originated (e.g., initiate, cancel, or modify a service) as well as changes to system parameters resulting from relatively minor office equipment and network rearrangements, (e.g., addition of a DSS to a growing shopping center previously served by individual lines). The service order change procedures are designed for quick response and for daily use.

(iii) *Maintenance and Traffic Measurements*. Maintenance and traffic measurements are made by the TN message switch as a result of trouble conditions and call processing. The data are used by the Dial Administration and Network Services group to engineer the system's memory, peripheral equipment, and transmission facilities and by the maintenance forces to evaluate system performance. When these data indicate the need for minor software or hardware reconfigurations, the service order change programs are utilized. If more extensive changes are required, reallocations are used.

(ii) *Audio Response Unit (ARU) Memory Loading*. One major administrative task which does not clearly fit into eitber of the two categories previously mentioned is the ARU memory loading. This function requires a program which will initially load the ARU memories from a cartridge tape associated with the TN message switch, as well as reload them after an ARU memory error or vocabulary change.

A factor affecting the organization of the administrative programs is that many of the tasks (with the exception of maintenance and traffic measurement programs) are executed infrequently compared to the call-processing programs. To allocate the 3A Processor main memory in an efficient manner, some administrative programs are stored in the cartridge tape and loaded into memory as needed. This use of the cartridge tape system to store the nonresident programs allows more main memory to be allocated to control blocks and message buffers which grow proportionately as the network expands.

Programs residing in the cartridge tape are called *nonresident* programs. Associated with these nonresident programs are some *resident* programs called "overlay monitors," which are equipped to accept teletypewriter (TTY) messages from the TTY handler programs and then take the necessary steps to bring the required administrative programs from the cartridge tape into a buffer area of memory. Once the particular program has been loaded, the input message data is processed and the required actions are taken. Control then passes back to the "overlay monitor" program which is then ready to accept a new message.

The various administrative functions are described in more detail in Sections 7.2 through 7.5.

7.2 Reallocation programs

7.2.1 General considerations

The TN system requires the capability to allocate and reclaim memory for any existing control block or table. This facility is necessary to accommodate major office equipment and customer growth, and general reorganizations of existing control blocks as new storage is added to the system. The service order change programs are not capable of dealing with such reorganizations because they are only equipped to handle minor additions to, deletions from, and changes to existing data structures.

As an example, whenever the number of polled terminal control blocks (TCBs) used approaches a certain percentage of the total number of blocks allocated, then, to insure enough blocks for future expansion, the number of blocks available must be increased. Such a reallocation would probably impact other memory blocks as well, resulting in a general reorganization of the entire memory in which some groups of blocks will either expand or contract.

Reallocations are not procedures to be performed regularly, but are intended to be used as needed to reflect the changing system capacity. Furthermore, the nature of the reallocation process is such that it should only be performed at a time of minimal network activity such as late night or weekends. New call processing is suspended, allowing the processor to reach a stable state while the blocks in memory are actually being changed. Because of the infrequency of reallocation, the programs required to implement this function reside on a cartridge tape.

The reallocation program can perform in three modes: Normal, Initial, or Top. In the Normal mode, the changes specified by the craftsperson are made to the currently existing data structure and data in the existing control blocks are preserved through reallocation. In the Initial mode, all previously existing allocations are destroyed and the memory is reconfigured from scratch. In the Top mode, a new system memory size may be specified without changing the number of control blocks allocated. In this case, the allocated blocks, which are stored at the top (high addresses) of memory are moved to the new top of memory.

7.2.2 Reallocation program organization

The reallocation programs consist of a resident input message handler and a nonresident master reallocation program. One function of the input message handler is simply to accept an input message from an operator and load the master reallocation program into memory. The master program responds to subsequent TTY input messages which define the new memory layout (Section 7.2.3 discusses the procedure further). This layout results in a new Master Block Array (MBA). The MBA is essentially a blueprint of the various types of memory blocks. An MBA is made up

of a series of contiguous nine-word blocks, each of which defines the layout in memory of one type of memory block or table. Figure 3 illustrates the type of information stored in the contiguous nine-word blocks.

Once a new MBA has been created, tbe reallocation portion of the master program uses it to restructure memory. During the course of the reallocation, this program also accepts TTY messages which verify the validity of the input data. When the master program is no longer required, a separate TTY message is used to deactivate it, i.e., it is erased from memory.

7.2.3 Reallocation procedure

When it is necessary to perform a reallocation, a list is prepared which contains all the data needed to construct a new Master Block Array. The craftsperson initiates the process by typing an input message from the maintenance TTY to start the procedure previously described. The existing MBA is copied into a work area of memory. Subsequent TTY input messages are used to change the various entries in this "scratch" MBA. When no further modifications are to be made, the craftsperson indicates this by requesting a printout of the essential elements of the new MBA to compare it with the original list of inputs. After this verification step, the craftsperson initiates a system slowdown. That is, the reallocation program informs the call processing programs not to undertake any new calls and to leave all current activity in a stable quiescent state. The duplex mode of operation in the 3A Processors is then suspended temporarily by the reallocation program by putting the standby processor in an out-of-service mode. Meanwhile, the on-line processor continues normal operations. The real reallocation process then begins in the off-line processor. The reallocation program accomplishes this by comparing the old MBA with the scratch MBA and then shifting and expanding the various control blocks and overwriting the original MBA in the off-line processor memory. When this step is completed, a processor switch occurs (the off-line processor becomes on-line), and an output message is typed to indicate that the reallocation has been performed in one store, which is now the on-line store. The craftsperson lets the system run for a while to ensure that it is operating normally. Then the new off-line main store is updated via another TTY message. Both main stores now contain the same information.

7.3 Service order change programs

As previously mentioned, service order changes are required to reflect updates in system memory to subscriber-originated changes as well as to changes in system parameters resulting from office equipment and network rearrangements and/or growth.

Service order changes are initiated from either the service order or maintenance teletypewriter. Each TTY input message covers a specific type of change activity and is associated with a particular type of control block. Typically, each message is executed by a different service order change subroutine. However, in some cases more than one service order change input is handled by a single subroutine. Since service order changes are not procedures that are performed often (compared to call processing and certain maintenance programs), these programs are nonresident and are loaded from the cartridge tape as needed.

7.3.1 Service order change program organization

All service order change input messages are routed to a resident service order change input message handler. This program functions as a monitor in that it decides (based on which input message was sent) which programs will be needed to execute the requested action. Once this has been determined, it initiates the retrieval of the appropriate non-resident program into memory. When this is done, control is transferred to the subroutine in the nonresident program which will actually respond to the service order change request. During this phase of processing, control is not passed back to the input message handler program until the requested action has been either completed or aborted due to some input error. In the case of a successful completion, the input message handler issues a completion message to the originating TTY. If an error was detected, it issues an error message with a code indicating the reason for the failure. The input message handler is then ready to accept a new input message.

7.3.2 Processor control and memory protection

Since service order changes modify protected system memory, steps have to be taken to insure system recovery in case of an initialization during a service order change (regardless of how the initialization was caused). This is accomplished by performing the changes in the on-line memory while the off-line memory is out of service (i.e., frozen in a previous state). This configuration is obtained at the beginning of the service order change session by typing a begin service order change input message. Then all the required messages can be typed. When the session is completed, an input message is used to restore the offline processor back to the standby mode. Doing this copies the on-line memory (containing the recent changes) into the off-line memory so that the two are again identical. If an initialization were to occur before the service order change was completed, control would be transferred to the off-line processor and all the changes for that session would be lost. This is a small price to pay for the advantage of not having to reload memory from the cartridge tape.

Another safety feature consists of a time-out if no new messages are typed within 5 minutes of the last one. This prevents the off-line processor from being locked in the out-of-service state for an excessive idle period. The time-out causes the off-line memory to be updated and the processor to be placed back in the standby mode. An output message is issued, alerting the craftsperson to the fact that this has occurred.

7.3.3 Service order change verification

Part of the service order change package contains a group of programs that can cause the contents of certain control blocks to be printed on the TTY. These programs fall under the general heading of "verifications." The TTY input messages which perform this function are handled similarly to the service order change messages (i.e., by the input message handler).

7.3.4 Description of service order change/verification message processing

The input message handler is an event-driven task. In other words, the program is entered whenever a TTY input message is directed to it. Upon entry into the program, an immediate check is made to determine which one of the following situations has occurred:

(*i*) A single line message was sent: In this case, a check is made to determine if a message was already in progress. If so, then the new message is rejected (an RL—Repeat Later—response is issued), and processing of the in-progress message is resumed. If another message was not in progress, the new one is accepted, the overlay buffer is seized, and the appropriate nonresident programs are retrieved. In addition, a PF—Printout Follows—is issued. Control is then transferred to the program which will actually process the input data. Control is passed back to the input message handler when the processing program has either completed or detected an error. The input message handler releases the overlay buffer, generates an appropriate TTY output message, and prepares for another input message.

(*ii*) First line of multiline message was sent: This situation is handled exactly as in (*i*) above, except that a PF is not issued unless an error message is to be printed. Successful completion means that the input message handler will accept the next line of the multiline message. In this case, the only response to the TTY is a carriage return and line feed which is a signal to the craftsperson to enter the next line. Meanwhile, the processing (nonresident) program is waiting for the next line of the message.

(*iii*) Middle line of a multiline message sent: This is handled by simply accepting the line and entering the processing program at the point left off by the previous line. Errors and successful completion are handled as in (*i*) above.

(*iv*) Last line of a multiline message sent: Again, this results in the line being accepted and entering the processing program at the point left off by the previous line. A PF is issued by the processing program. Control then reverts back to the input message handler which issues either an error message or a completion message. At this point, the overlay buffer is released and the input message handler waits for the next message.

7.4 Maintenance and traffic measurement programs

Maintenance and traffic measurements are intended to provide (*i*) a means of monitoring message flow through the system, in terms of completed messages and ineffective attempts, (*ii*) usage of resources in the system, and (*iii*) trouble information about resources in the system. The data are intended to help engineer changes to the TN network for growth and optimization of existing equipment given the existing traffic requirements, and to help localize and identify faults affecting quality of service. Traffic data are put out through a dedicated TTY facility on assigned quarterly, hourly, and daily schedules, or on demand. Various combinations of the three basic types of measurements (peg counts, usage, and overflow) are performed in the traffic measurements. Maintenance measurements are printed on a daily basis, or on demand on the maintenance teletypewriter; they deal primarily with equipment outages and transient faults.

The maintenance and traffic measurement programs are divided into scanning tasks and reporting tasks. The scanning tasks are activated every 2, 10, or 100 seconds; their main function is to accumulate counts of various kinds. The reporting tasks print the maintenance and traffic reports.

The quarterly, hourly, and daily traffic reports and the maintenance reports all have different formats.

Traffic reports are printed according to a schedule selected by the telephone company and inputted via teletypewriter facilities. This schedule is stored in memory in the Traffic Work Table.

7.5 ARU loading

The TN message switch utilizes two ARUs, each independently serving 76 dial-in ports. The ARU speech memory is a semiconductor RAM which is loaded from a special magnetic tape cartridge through the message switch. This tape contains all the speech segments to be loaded into the ARU and tables to be loaded into the 3A Processor memory which contain the ARU speech segment addresses.

The ARU loading is controlled by a nonresident program which is manually activated depending on whether the ARU is to be loaded due to:

(*i*) An initial start-up procedure.
(*ii*) Recovery from a fault condition (e.g., power failure).

(*iii*) A vocabulary update.

Only one ARU is loaded at a time, and all dial-in ports served by the ARU are busied out during the loading process which takes approximately 10 minutes.

VIII. SUMMARY

The foregoing discussion has provided the general organization and structure of the operational TN software, an explanation of the functional tasks performed by the various programs, and a description of the call processing, maintenance, and administrative functions. The author has attempted to provide insight into the techniques and considerations used in the development of operational TN programs.

IX. ACKNOWLEDGMENTS

The design and implementation of the Extended Operating System represent the contributions of many individuals at Bell Laboratories. In particular, J. H. Campbell, C. H. Elmendorf, J. A. Lindner, and D. C. Plisch provided valuable support during the course of the TN software development. The operational TN software was developed by S. G. Anderson, L. R. Beaumont, D. G. Chirieleison, K. L. Cohen, P. J. Deceglie, P. J. Kreider, D. R. Johnson, F. S. Menzel, T. P. McLoughlin, W. E. Omohundro, and R. E. Reid. This article is based on their dedicated effort.

REFERENCES

1. J. A. Harr, E. S. Hoover, and R. B. Smith, "Organization of the No. 1 ESS Stored Program," B.S.T.J., *43*, No. 5, Part 1 (September 1964), pp. 1923–1959.
2. S. E. Madnich and J. J. Donovan, *Operating Systems*, New York McGraw-Hill, 1974.
3. "Transaction Network System Description," Bell System Data Communications Technical Reference, PUB 41024.
4. "Transaction Network Service Dial-in Interface Specification," Bell System Data Communications Technical Reference, PUB 41026.
5. "Transaction Network Service Polled Interface Specification," Bell System Data Communications Technical Reference, PUB 41025.
6. "Transaction Network Service Synchronous Interface Specification," Bell System Data Communications Technical Reference, PUB 41027.
7. H. J. Bruseher, G. E. Fessler, D. W. Huffman, P. J. Kennedy, and E. Nussbaum, "Administration and Maintenance Plan for No. 2 ESS," B.S.T.J., *48*, No. 8 (October 1969), pp. 2765–2815.